

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



**MÔN HỌC: VI XỬ LÝ - VI ĐIỀU KHIỂN (TN) (CO3010)**

---

**LAB 2 - TIMER INTERRUPT AND LED SCANNING**

---

**Giảng viên hướng dẫn:** Lê Trọng Nhân  
Tôn Huỳnh Long

**Lớp:** L02

**Sinh viên thực hiện:** Võ Thị Ánh Tuyết - 2213826



## Contents

1	Exercise 1	2
2	Exercise 2	4
3	Exercise 3	4
4	Exercise 4	6
5	Exercise 5	6
6	Exercise 6	6
7	Exercise 7	7
8	Exercise 8	8
9	Exercise 9	9
10	Exercise 10	12
11	Link	13

## 1 Exercise 1

**Report 1:** Capture your schematic from Proteus and show in the report

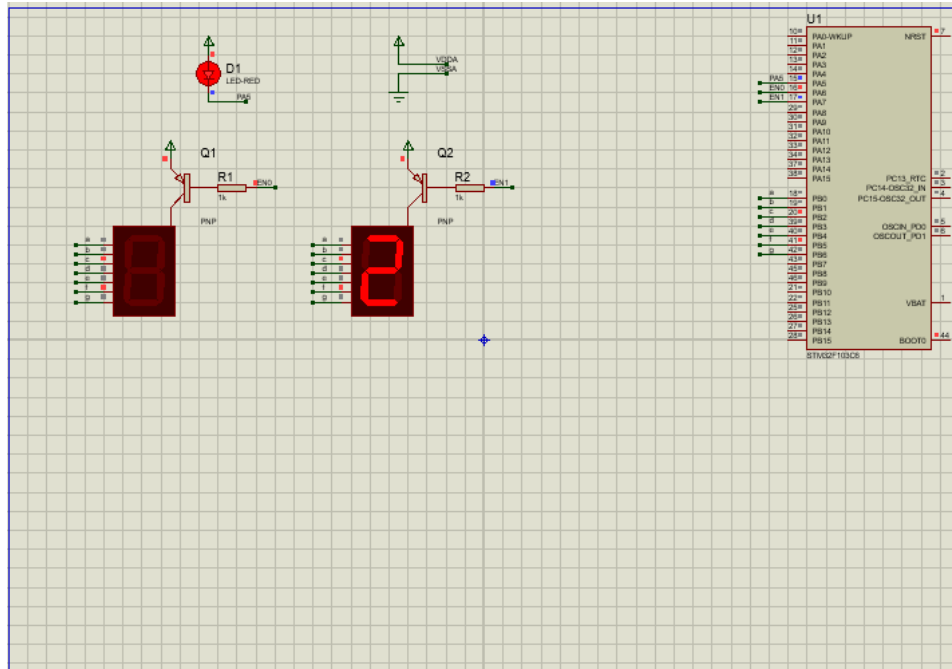


Figure 1: Exercise 1

**Report 2:** Present your source code in the HAL\_TIM\_PeriodElapsedCallback function

```
1 int num = 1;
2 void HAL_TIM_PeriodElapsedCallback ( TIM_HandleTypeDef * htim )
3 {
4     if (timer_flag[0] == 1)
5     {
6         HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
7         switch(num){
8             case 1 :
9                 HAL_GPIO_WritePin ( EN0_GPIO_Port , EN0_Pin ,
10                 GPIO_PIN_RESET ) ;
11                 HAL_GPIO_WritePin ( EN1_GPIO_Port , EN1_Pin ,
12                 GPIO_PIN_SET ) ;
13                 display7SEG(1);
14                 num = 2;
15                 break;
```

```
14         case 2 :  
15             HAL_GPIO_WritePin ( EN0_GPIO_Port , EN0_Pin ,  
GPIO_PIN_SET ) ;  
16             HAL_GPIO_WritePin ( EN1_GPIO_Port , EN1_Pin ,  
GPIO_PIN_RESET ) ;  
17             display7SEG(2);  
18             num = 1;  
19             break;  
20         default:  
21             break;  
22     }  
23     set_timer(0, 500);  
24 }  
25 timer_run();  
26 }
```

Listing 1: Source code in HAL\_TIM\_Period Elapsed Callback function.

**Short question:**  $F = 1/10 \times 10^{-3} = 100 \text{ (Hz)}$

## 2 Exercise 2

**Report 1:** Capture your schematic from Proteus and show in the report. **Report**

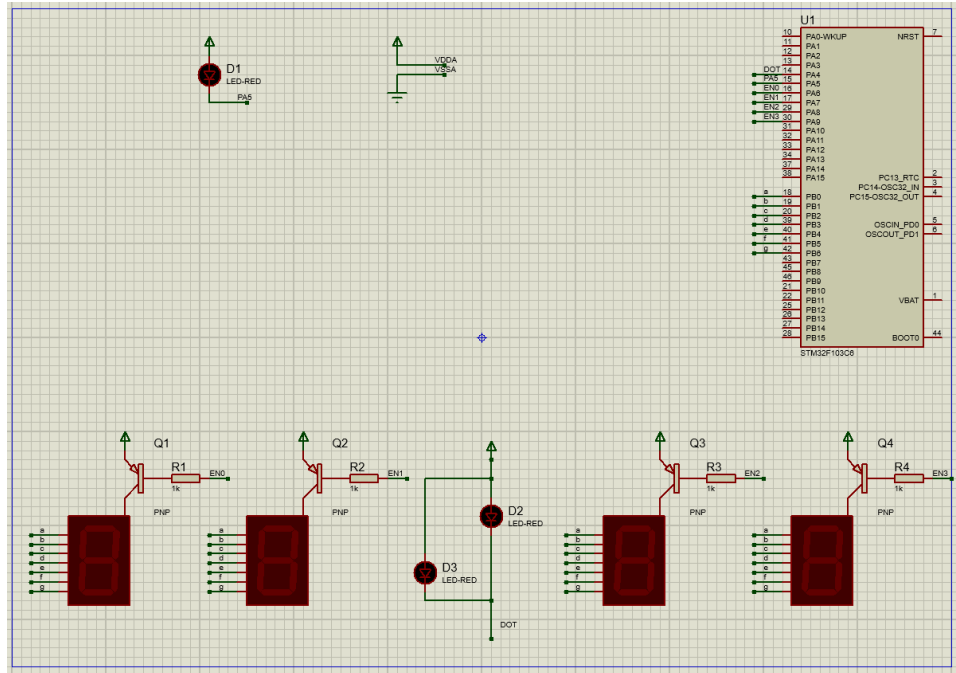


Figure 2: Exercise 2

**2:** Present your source code in the HAL\_TIM\_Period Elapsed Callback function.

```
1 void HAL_TIM_PeriodElapsedCallback ( TIM_HandleTypeDef * htim )
2 {
3     timer_run();
4 }
```

Listing 2: source code in the HAL\_TIM\_Period Elapsed Callback function

**Short question:**  $F = 1/10 \times 10^{-3} = 100 \text{ (Hz)}$

## 3 Exercise 3

**Report 1:** Present the source code of the update7SEG function.

```
1 void update7SEG(int index)
2 {
3     switch(index)
4     {
```

```
5  case 0:
6      HAL_GPIO_WritePin(EN0_GPIO_Port, EN0_Pin, RESET);
7      HAL_GPIO_WritePin(EN1_GPIO_Port, EN1_Pin, SET);
8      HAL_GPIO_WritePin(EN2_GPIO_Port, EN2_Pin, SET);
9      HAL_GPIO_WritePin(EN3_GPIO_Port, EN3_Pin, SET);
10     display7SEG(led_bufer[index]);
11     break;
12 case 1:
13     HAL_GPIO_WritePin(EN0_GPIO_Port, EN0_Pin, SET);
14     HAL_GPIO_WritePin(EN1_GPIO_Port, EN1_Pin, RESET);
15     HAL_GPIO_WritePin(EN2_GPIO_Port, EN2_Pin, SET);
16     HAL_GPIO_WritePin(EN3_GPIO_Port, EN3_Pin, SET);
17     display7SEG(led_bufer[index]);
18     break;
19 case 2:
20     HAL_GPIO_WritePin(EN0_GPIO_Port, EN0_Pin, SET);
21     HAL_GPIO_WritePin(EN1_GPIO_Port, EN1_Pin, SET);
22     HAL_GPIO_WritePin(EN2_GPIO_Port, EN2_Pin, RESET);
23     HAL_GPIO_WritePin(EN3_GPIO_Port, EN3_Pin, SET);
24     display7SEG(led_bufer[index]);
25     break;
26 case 3:
27     HAL_GPIO_WritePin(EN0_GPIO_Port, EN0_Pin, SET);
28     HAL_GPIO_WritePin(EN1_GPIO_Port, EN1_Pin, SET);
29     HAL_GPIO_WritePin(EN2_GPIO_Port, EN2_Pin, SET);
30     HAL_GPIO_WritePin(EN3_GPIO_Port, EN3_Pin, RESET);
31     display7SEG(led_bufer[index]);
32     index = 0;
33     break;
34 default:
35     break;
36 }
37 }
```

Listing 3: Source code of the update7SEG function

**Report 2:** Present your source code in the HAL\_TIM\_Period Elapsed Callback function.

```
1 void HAL_TIM_PeriodElapsedCallback ( TIM_HandleTypeDef * htim )
2 {
```

```
3   timer_run();  
4 }
```

Listing 4: source code in the HAL\_TIM\_Period Elapsed Callback function

## 4 Exercise 4

**Report 1:** Already presented in previous exercise

## 5 Exercise 5

**Report 1:** Present the source code in the **updateClockBuffer** function.

```
1   int hour = 15 , minute = 8 , second = 50;  
2  
3   void updateClockBuffer()  
4   {  
5       led_bufer[0] = hour / 10;  
6       led_bufer[1] = hour % 10;  
7       led_bufer[2] = minute / 10;  
8       led_bufer[3] = minute % 10;  
9   }
```

Listing 5: Source code in the updateClockBuffer function

## 6 Exercise 6

**Report 1:** if in line 1 of the code above is miss, what happens after that and why?  
If in line 1 of the code above is miss, the led is not going to blink because the value of timer\_flag[0] is not set to 0 and never satisfy the condition of if statement in while loop.

**Report 2:** : if in line 1 of the code above is changed to setTimer0(1), what happens after that and why?

If in line 1 of the code above is changed to setTimer0(1), LED is dim because the flashing frequency is too high.

**Report 2:** : if in line 1 of the code above is changed to setTimer0(10), what is changed compared to 2 first questions and why?

If in line 1 of the code above is changed to `setTimer0(10)`, LED blinks slower than in case 2 but faster than in case 1.

## 7 Exercise 7

**Report 1:** Present your source code in the while loop on main function

```
1  while (1)
2  {
3      if(timer_flag[0] == 1 )
4      {
5          HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_4);
6          set_timer(0, 1000);
7      }
8      if(timer_flag[1] == 1)
9      {
10         second ++;
11         if ( second >= 60)
12         {
13             second = 0;
14             minute ++;
15         }
16         if( minute >= 60)
17         {
18             minute = 0;
19             hour ++;
20         }
21         if( hour >=24)
22         {
23             hour = 0;
24         }
25         set_timer(1, 1000);
26     }
27
28     updateClockBuffer () ;
29 }
```

Listing 6: Source code in the while loop function



## 8 Exercise 8

**Report 1:** Present your source code in the the main function.

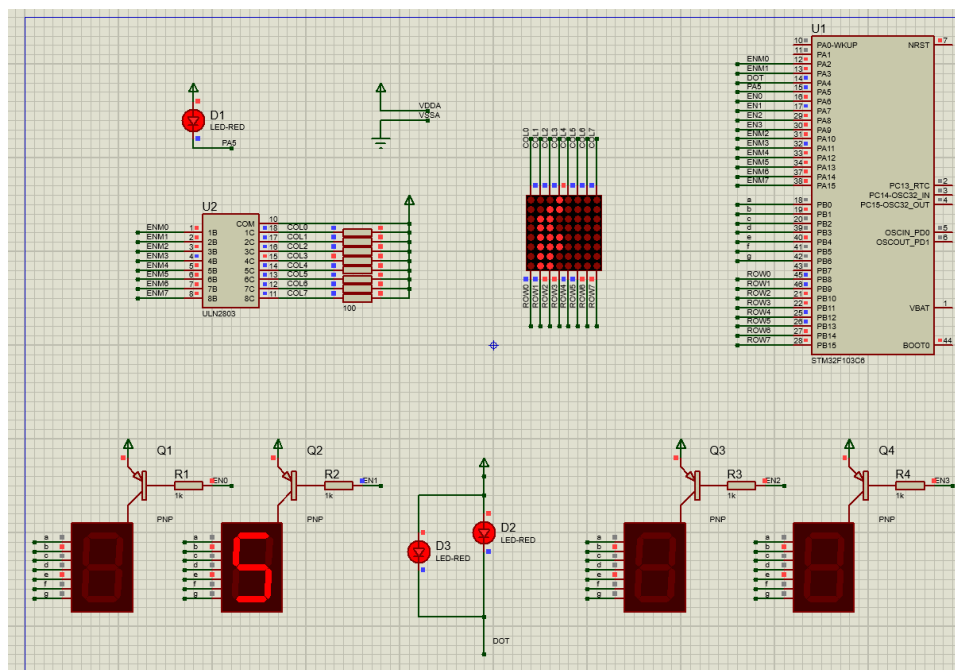
```
1  set_timer(0, 1);
2  set_timer(1, 2);
3  set_timer(2, 3);
4  int index_led = 0;
5
6  while (1)
7  {
8      if(timer_flag[0] == 1 )
9      {
10         HAL_GPIO_TogglePin(GPIOA , GPIO_PIN_4);
11         set_timer(0, 1000);
12     }
13     if(timer_flag[1] == 1)
14     {
15         second ++;
16         if ( second >= 60)
17         {
18             second = 0;
19             minute ++;
20         }
21         if( minute >= 60)
22         {
23             minute = 0;
24             hour ++;
25         }
26         if( hour >=24)
27         {
28             hour = 0;
29         }
30         set_timer(1, 1000);
31     }
32     if (timer_flag[2] == 1)
33     {
34         updateClockBuffer () ;
35         update7SEG(index_led++);
36         if(index_led > 3) index_led = 0;
```

```
37         set_timer(2, 100);
38     }
```

Listing 7: Source code in the the main function

## 9 Exercise 9

**Report 1:** Present the schematic of your system by capturing the screen in Proteus.



**Figure 3:** *Exercise 9*

**Report 2:** Implement the function, `updateLEDMatrix(int index)`, which is similarly to 4 seven led segments

```
1 void updateMatrix (int index)
2 {
3     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, SET);
4     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, SET);
5     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, SET);
6     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, SET);
7     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, SET);
8     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, SET);
```

```
9  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, SET);
10 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, SET);
11
12 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, SET);
13 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, SET);
14 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, SET);
15 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, SET);
16 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, SET);
17 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, SET);
18 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, SET);
19 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, SET);
20 switch ( index )
21 {
22     case 0:
23         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, RESET);
24         break ;
25     case 1:
26         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, RESET);
27
28         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, RESET);
29         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, RESET);
30         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, RESET);
31         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, RESET);
32         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, RESET);
33         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, RESET);
34         break ;
35     case 2:
36         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, RESET);
37
38         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, RESET);
39         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, RESET);
40         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, RESET);
41         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, RESET);
42         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, RESET);
43         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, RESET);
44         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, RESET);
45         break ;
46     case 3:
47         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, RESET);
```

```
48
49     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, RESET);
50     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, RESET);
51     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, RESET);
52     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, RESET);
53     break ;
54     case 4:
55         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, RESET);
56
57         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, RESET);
58         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, RESET);
59         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, RESET);
60         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, RESET);
61     break ;
62     case 5:
63         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, RESET);
64
65         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, RESET);
66         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, RESET);
67         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, RESET);
68         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, RESET);
69         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, RESET);
70         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, RESET);
71         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, RESET);
72     break ;
73     case 6:
74         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, RESET);
75
76         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, RESET);
77         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, RESET);
78         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, RESET);
79         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, RESET);
80         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, RESET);
81         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, RESET);
82     break ;
83     case 7:
84         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, RESET);
85     break ;
86     default :
```

```
87     break ;  
88 }  
89 }
```

Listing 8: Source code of the updateLEDMatrix(int index) function

## 10 Exercise 10

**Report 1:** Briefly describe your solution and present your source code in the report

```
1  initScrollingText();  
2  set_timer(0, 1);  
3  set_timer(1, 2);  
4  set_timer(2, 3);  
5  set_timer(3, 4);  
6  
7  int index_matrix = 0;  
8  while(1)  
9  {  
10     if(timer_flag[0] == 1)  
11     {  
12         HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_4);  
13         set_timer(0, 1000);  
14     }  
15     if(timer_flag[1] == 1)  
16     {  
17         updateMatrix(index_matrix++);  
18         if(index_matrix > 7) index_matrix = 0;  
19         set_timer(1, 100);  
20     }  
21     if(timer_flag[2] == 1)  
22     {  
23         shiftDisplayLeft();  
24         updateMatrixBuffer();  
25         set_timer(2, 200);  
26     }  
27 }
```

Listing 9: e an animation on LED matrix



## 11 Link

For more details. please refer to this source code [Link Github](#)